1-1-2003

Six d.o.f Haptic Rendered Simulation of the Peg-in-Hole Assembly

Shen Dong
University of Wollongong, shen@uow.edu.au

Fazel Naghdy
University of Wollongong, fazel@uow.edu.au

Yan Chen
University of Wollongong

Six d.o.f Haptic Rendered Simulation of the Peg-in-Hole Assembly

Abstract

A new paradigm for programming of robotics manipulator to perform complex constrained motion tasks is being studied. The teaching of the manipulation skills to the machine starts by demonstmting those skills in a haptic -rendered virtual environment. Position and contact force and torque data generated in the virtual environment combined with a priori knowledge about the task is used to identify and learn the skills in the newly demonstrated tasks and then to reproduce them in the robotics system. The peg-in-hole insertion problem is used as a case study. In the ftrst implementation of the virtual manipulahon envlfonment, a three degree-offreedom (3DOF) haptic device called PHANToM Premium 1.0 and its accompanying software, GHOST were used to construct the virtual manipulation. The developed system proved quite stable when the peg-in-hole insertion was performed using 3DOF Phantom. However, employment of a six degree-of-freedom PHANToM Premium 1.5 destabilised the process and introduced strong oscillations. The work conducted to stabilise this system is reported. Three new algorithms including modilled PointShell, modifted TriPolyMesh and dual-gstCylinder developed and applied to the virtual assembly are explained. The major differences between the new algorithms and previous methods are highlighted. The progress up-to-date is provided.

Keywords

hole, peg, assembly, rendered, haptic, f, o, simulation, six

Disciplines

Physical Sciences and Mathematics

Publication Details

S. Dong, F. Naghdy & Y. Chen, "Six d.o.f Haptic Rendered Simulation of the Peg-in-Hole Assembly," in 9th International Conference on Manufacturing Excellence 2003, 2003,

Six d.o.f Haptic Rendered Simulation of the Peg-in-Hole Assembly

S. Dong, F. Naghdy, Y. Chen

School of Electrical, Computer and Telecommunications Engineering,
University of Wollongong, Australia
Email: f.naghdy@uow.edu.au

Keywords
haptics, peg-in-hole insertion, virtual manipulation, virtual environment.

Abstract
A new paradigm for programming of robotics manipulator to perform complex constrained motion tasks is being studied. The teaching of the manipulation skills to the machine starts by demonstrating those skills in a haptic-rendered virtual environment. Position and contact force and torque data generated in the virtual environment combined with a priori knowledge about the task is used to identify and learn the skills in the newly demonstrated tasks and then to reproduce them in the robotics system. The peg-in-hole insertion problem is used as a case study. In the first implementation of the virtual manipulation environment, a three degree-of-freedom (3DOF) haptic device called PHANToM Premium 1.0 and its accompanying software, GHOST were used to construct the virtual manipulation. The developed system proved quite stable when the peg-in-hole insertion was performed using 3DOF Phantom. However, employment of a six degree-of-freedom PHANToM Premium 1.5 destabilised the process and introduced strong oscillations. The work conducted to stabilise this system is reported. Three new algorithms including modified PointShell, modified TriPolyMesh and dual-gstCylinder developed and applied to the virtual assembly are explained. The major differences between the new algorithms and previous methods are highlighted. The progress up-to-date is provided.

1 Introduction

A new paradigm for programming of robotics manipulator to perform complex constrained motion tasks is being studied. The teaching of the manipulation skills to the machine starts by demonstrating those skills in a haptic-rendered virtual environment.

A Haptic interface, or force feedback device has the potential to increase the quality of human-computer interaction by accommodating the sense of touch in computer simulation. It provides an attractive augmentation to visual display and to enhance the level of immersion in a virtual world. Haptic interface has been effectively used in a number of applications including surgical procedures training, virtual prototyping, control panel operations, hostile work environments and manipulation of materials.

The peg-in-hole insertion process was used as a platform to study the concept. The insertion of a peg in a hole is often taken as a standard assembly problem, as it concisely represents a constrained motion force sensitive manufacturing task with all the attendant issues of jamming, tight clearances, and the need for quick insertion times, reliably, etc.

In the developed system, position and contact force and torque data generated in the virtual environment combined with a priori knowledge about the task is used to identify and learn the skills in the newly demonstrated tasks and then to reproduce them in the robotics system.
In the first implementation of the virtual manipulation environment, a three degree-of-freedom (3DOF) haptic device called PHANToM Premium 1.0 and its accompanying software, GHOST were used to construct the virtual manipulation.
The graphics model of the assembly [1] was constructed using OpenGL, whereas its physical model and the force/torque vectors generated in the virtual manipulation environment were modelled based on two different approaches of PointShell and TriPolyMesh [2]. The developed system proved quite stable when the peg-in-hole insertion was performed using 3DOF Phantom.
In complex applications, where simulation of arbitrary object to object interaction is required, a six degree-of-freedom (6DOF) haptic device, can be a more effective tool. It provides torque feedback in addition to force display within a large translation and rotational range of motion, and provides the user with the much needed dexterity to feel, explore and maneuver around other objects in the virtual environment. In order to improve the performance of the developed system, the haptic device was upgraded to a six degree-of-freedom PHANTOM Premium 1.5. This resulted in strong oscillation occurring during virtual peg-in-hole insertion, preventing a successful insertion of the peg into the hole.
Further investigation of the problem revealed that PointShell and TriPolyMesh algorithms used in the model were not sufficiently accurate for operation with a 6 DOF haptic device.
In order to stabilize the virtual peg-in-hole insertion with tight fit for 6DOF haptic device, three new algorithms have been developed and applied to the physical model of the process. They include modified PointShell algorithm, modified TriPolyMesh algorithm and dual-gstCylinder algorithm.
The rest of the paper is organized as follows. Section 2 briefly provides a review of the previous work related to this project. The methodologies developed for the three degree-of-freedom (3DOF) haptic device are covered in Section 3. The algorithms developed to stabilize the virtual peg-in-hole insertion with a 6DOF haptic device are described in Section 4. Finally, concluding remarks are given in Section 5.

2. Haptic Rendering

Haptics is relatively a recent enhancement to virtual environments allowing users to "touch" and "feel" the simulated objects they interact with [3]. The international scientific vocabulary defines haptic as relating to or based on the sense of touch or characterized by a predilection for the sense of touch [7]. Haptic rendering is the process of applying forces representing a virtual environment to the user of a haptic interface, through a force feedback device. The rendering process consists of using information collected from the virtual environment, evaluating the forces and torques to be reacted at given a position, velocity, etc. of the operational point of a haptic interface. The operational point, or points, is the physical location on the haptic interface where position, velocity, acceleration, and sometimes force, are measured [8]. In order to display a virtual environment, the flowing problems must be addressed [9]:

- Finding the point of contact: This is the problem of collision detection, which becomes more difficult and computationally expensive as the model of the virtual environment becomes more complex.
- Generation of contact forces: This creates the "feel" of the object. Contact forces can represent the stiffness of the object, damping, friction, surface texture, etc.

- Dynamics of the virtual environment: Objects manipulated in a virtual environment can collide with each other and move in a complicated way.
- Computational rate: Computational rate must be high, around 1 kHz or higher, and the latency must be low. Inappropriate values of both these variables can cause hard surfaces in the virtual environment to feel soft as well as creating system instabilities.

Haptic rendered object models have primarily been simple geometries such as polyhedrons, cylinders, spheres, and other basic shapes. In the most basic and commonly used haptic rendering algorithm, the virtual environment is modeled as intersecting planes, with normal and tangential force vectors as defined above. Basic shapes, such as polyhedrons, cylinders, and spheres, can be used to model objects, but there will still be limitations. More sophisticated haptic rendering methods have been developed to deal with these problems.

General rendering allows for the stable display of dynamic environments with interacting objects. The most basic model used in haptic rendering is polygon, as it provides simple methods to calculate intersection of the objects and interaction forces and torques. Using an implicit function is another way to model an object. Such a function is a compact mathematical inside-outside description of a surface, which can be easily used to calculate collisions between the object and points in space. NURBS (Non-Uniform Rational B-Splines) and other model typical used in graphics of CAD environments can also be used as models for haptic rendering. Surface properties of virtual objects may be displayed using texture algorithms [10].

The PHANToM family of haptic devices, manufactured by SensAble Technologies (Boston, MA), is currently the most widely used force feedback interface on the market. The PHANToM has 3 or 6 Degrees of Freedom (DOF) and it can provide wrist motion up to shoulder motion depending on the model. A PHANToM can produce a maximum transient force of up to 22 N, and a sustained force of 3 N. In six (DOF) device, the maximum torques generated is 670 mNm, being produces by actuators placed in the handle. Continues torques is 104 mNm [6]. The characteristics of the PHANToM make it well suited for point interaction, for example, operated by a single virtual finger, a pencil or a peg.

The GHOST® SDK is the software package, which accompanies PHANToM. It is developed by SensAble Technologies for the PHANToM haptic interface, and consists of a C++ library of objects and methods used for developing interactive, three-dimensional, touch-enabled environments [5]. This software package handles complex computations and allows developers to specify object geometry, properties, and global haptic effects, using a haptic scene graph. It also allows programmers to deal with simple, high-level objects and physical properties like locations, mass, friction and stiffness [4]. GHOST automatically computes the interaction forces between a haptic point, and objects within the virtual environment. It can also simulate object compliance, friction, springs, impulses and vibrations. The GHOST SDK provides an abstraction that allows application developers to concentrate on the generation of haptic scenes, manipulation of the properties of the scene and objects within the scene, and control of the resulting effects on or by one or more haptic interaction devices. Thus, application developers do not need to be concerned with low-level force and device issues [5]. Ghost SDK does not generate graphics scenes, and a different tool such as OpenGL is needed.

3. Haptic Rendering for 3 DOF Device

In the model developed for 3 DOF haptic device, the virtual peg is coupled with the phantom (i.e. the manipulation point) through a spring-damper system. The peg is a dynamic rigid object in the virtual environment. The force reacted to the peg are transferred to PHANTOM Premium 1.0 through the spring damper system. The virtual hole is static in the virtual environment while the peg can be moved and rotated (Fig. 1) [2].

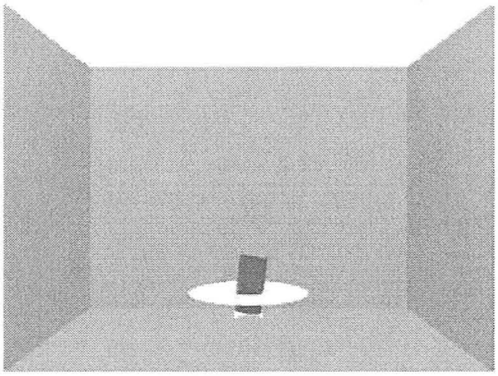


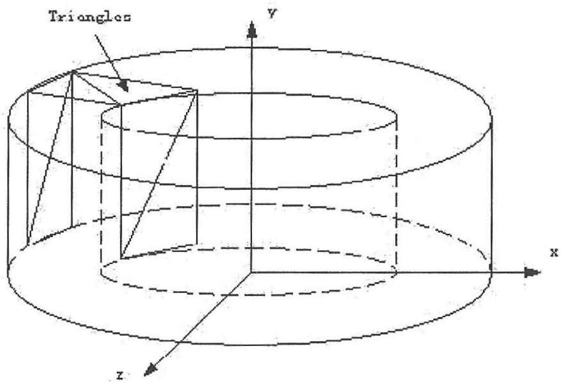
Figure. 1 3DOF Peg-in-hole Insertion Virtual Environment

The haptic rendered model of the peg-in-hole insertion generating force data is constructed using the TriPolyMesh and PointShell methods. The triangle polygon mesh (TriPolyMesh) method is used to construct the haptic model of the peg and hole. The surfaces of the virtual peg and hole are formed by rotating the triangles around the coordinate center (Fig. 2) [2]. In the PointShell method an object represented as a collection of a group of important points on their surface. A surface normal vector pointing inwards is assigned to every point on the PointShell to provide the Coulomb force direction [11]. Fig. 3(a) illustrates the normal vectors of a PointShell. In the PointShell developed for the peg-in-hole insertion, the directions of the vectors assigned to singular points are not pre-determined as they depend on the normal of the contact surface (Fig. 3(b)) [2]. The directions are assigned when the peg and hole are in contact.

The developed haptic rendered model proved quite stable when the peg-in-hole insertion was performed by the 3DOF Phantom. However, when at a later stage, the haptic device was upgraded to a six degree-of-freedom device, strong oscillation occurred during virtual peg-in-hole insertion and consequently the simulation could not be carried out successfully (Fig. 4). Further investigation of the problem revealed that PointShell and TriPolyMesh algorithms used in the model were not sufficiently accurate for operation with a 6 DOF haptic device.

4. Haptic Rendered Model for 6 DOF Device

In order to stabilize the virtual peg-in-hole insertion with tight fit for 6DOF haptic device, three new algorithms have been developed and applied to the physical model of the process. They include modified PointShell algorithm, modified TriPolyMesh algorithm and dual-gstCylinder algorithm.



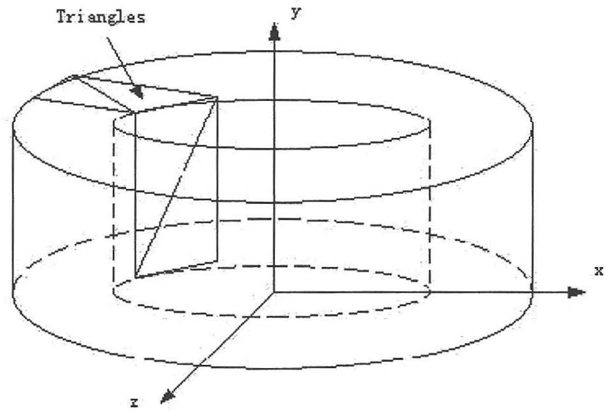


Figure. 2 TriPolyMesh Method

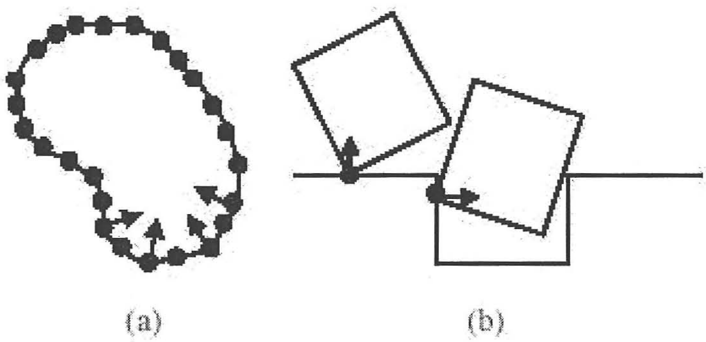


Figure. 3 PointShell Method

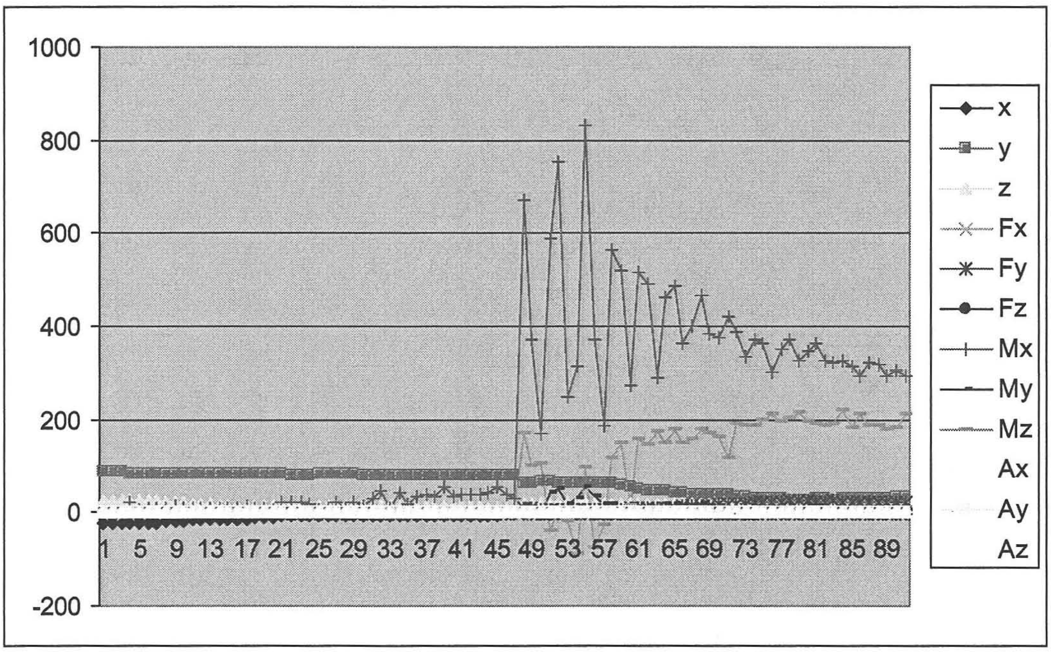


Figure. 4 strong oscillation indicated by high torques

In modified TriPolyMesh algorithm, the haptic hole is constructed by triangle polygon mesh algorithm and the haptic peg is gstCylinder, which is a cylinder shape class defined by GHOST SDK representing geometric primitive cylinder. The inside, outside and top surfaces of the hole are formed by rotating triangle polygons around y-axis as shown in Fig. 5. The gstPoints, one of a variety of data types defined and used in the GHOST API, are added to the vertex of each triangle polygon, represent Cartesian three dimensional point class. A group of gstPoints is also added to the end edges of the peg. The gstPoint generates contact force and torque data when it intersects with the body of peg. Similar to the previous model, the direction of the vectors assigned to each gstPoint is not predetermined in advance as it depends on the normal of the contact surface. This is determined according to the contact position when the peg and the hole come in contact. The gstPoints also play an important role in approximately removing the gaps produced when a polygon is used instead of a circle, as shown in Fig. 6.

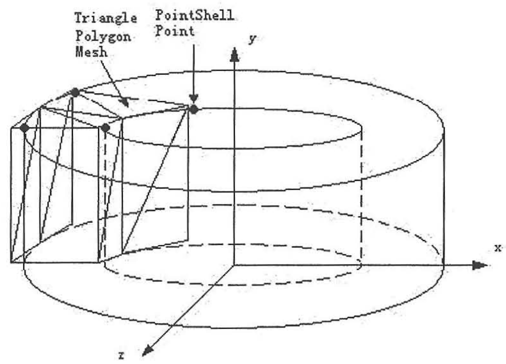


Figure. 5 Triangle Polygon Mesh

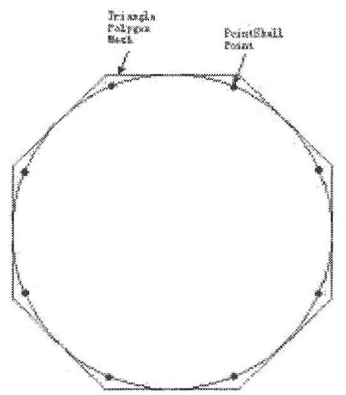


Figure. 6 Modified TriPolyMesh Method

In dual-gstCylinder algorithm, the hole is constructed using two gstCylinders rather then triangle polygons, forming the inner and outer surfaces of the hole (Fig. 7). This approach is simple and constructed model conforms well to the shape of the hole and hence there is no in inaccuracy in the model. The approach also offers a simple technique for the construction of the haptic rendered model. This haptic rendered model is a real cylinder model, so there is no accurate problem according to the approximate cylinder created by triangle polygon mesh algorithm. The gstPoints are introduced to edge of the hole and the two ends of the peg.

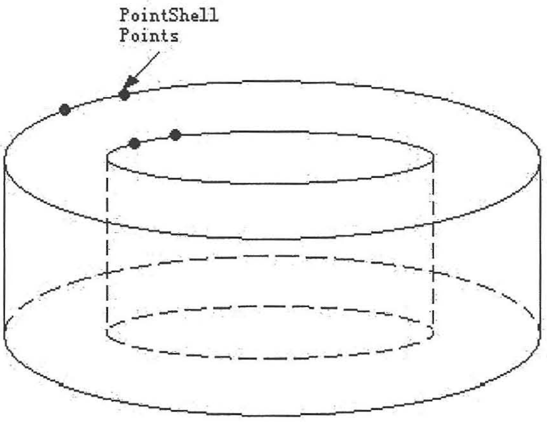


Figure. 7 Dual-gstCylinder Method

In the PointShell algorithm, the hole is defined as a dynamic object created by a group of gstPoints as shown in Fig. 8. The peg is also constructed by gstCylinder. The direction of the force generated at each gstPoint is normal to the surface of the hole at each point. The gstPoints on the hole prevents the virtual peg to penetrate into the inner surface of the hole.

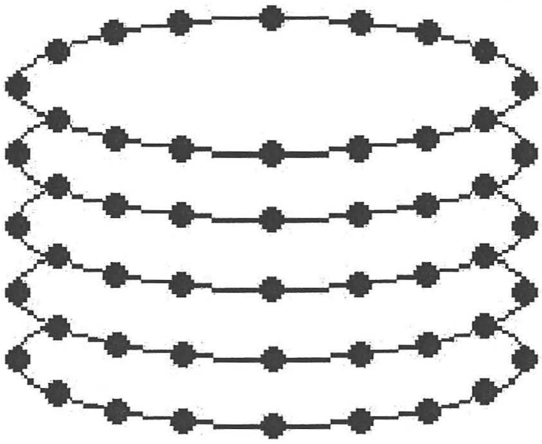


Figure. 8 PointShell Method

The force generated at each point is the sum of the Coulomb force and the friction force exerted at that point, as shown in Fig. 9 [1].

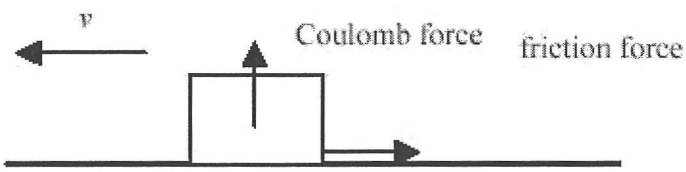


Figure. 9 Coulomb force and friction force

The direction of the Coulomb force is perpendicular to the contact surface and points to the moving object. The magnitude of the Coulomb force generated at each point is calculated by

$$fc = k \times d + c \times ad + b \times v$$

where

d is the depth of the point in the contacting static object

ad is the accumulated depth during a continuous contact between the point and the static object

v is the velocity of the object and is calculated by the current Depth minus the last Depth divided by the sampling time

k is the stiffness coefficient

b is the damping coefficient

c is the coefficient for the accumulated depth

The torque generated at each point is calculated by

$$t = fc \times D$$

where

D is the distance from the contacting point to the rotating center of the object

The direction of the friction force is along the contact surface and opposite to the moving direction. The magnitude of the friction force generated at each point is calculated by

$$f = \sigma \times z$$

where

z is the strain describing micro movements between the two objects, which is not allowed to exceed a small value called the breakaway distance z_{max}.

s is the stiffness relating force to strain Assume x_i is a point fixed on the moving object, and y_i is an adhesion point on the static object as shown in Fig. 10 [1].

The following relationship is used to calculate z_i by

$$z_i = x_i - y_i$$

$$y_i = x_i \mp z_{\max}, if |x_i - y_{i-1}| > z_{\max}$$

$$y_i = y_{i-1}, otherwise$$

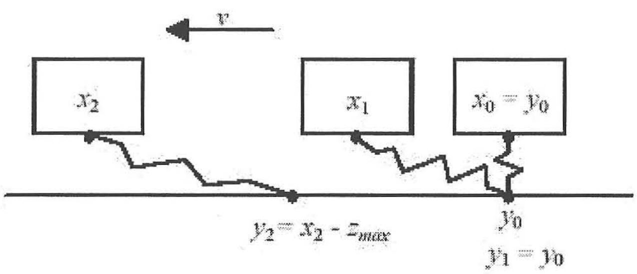


Figure. 10 The definition of the strain z

5. Results

The developed algorithms were applied to the virtual peg-in-hole insertion process. The process proved stable and no jamming was observed when the assembly was performed in the virtual environment with a 6DOF PHANToM Premium 1.5.

Some of the experimental results are illustrated in Figures 11-13. The variation of 11 normalized series of x, y, z, fx, fy, fz, Mx, My, Mz, Ax, Ay and Az are illustrated in these diagrams, where

- x, y, z are position of PHANToM or peg in world coordinates [millimeters].
- fx, fy, fz are reaction force in world reference frame from PHANToM [Newtons],
- Mx, My, Mz are reaction torque in world reference frame from PHANToM [Newton*millimeters].
- Ax, Ay, Az are equivalent rotation of current rotation matrix (orientation) based on successive rotations around x, y, z axes. Angles are in radians and right hand rule is used.

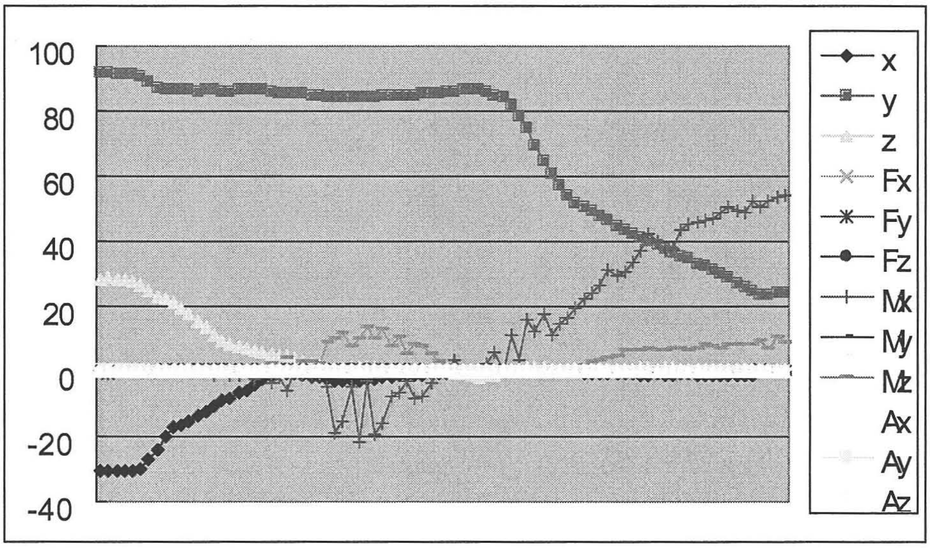


Figure 11 Modified TriPolyMesh Method

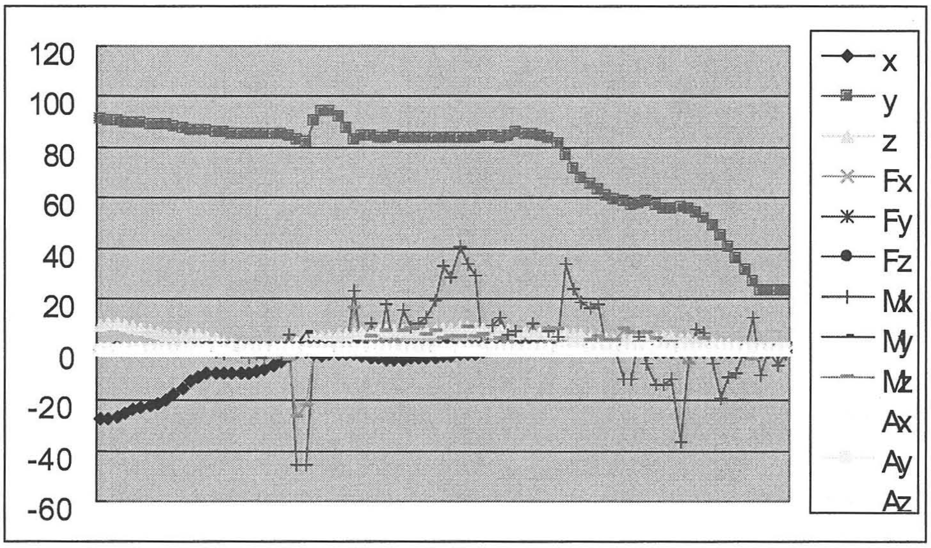


Figure. 12 Dual-gstCylinder Method Method

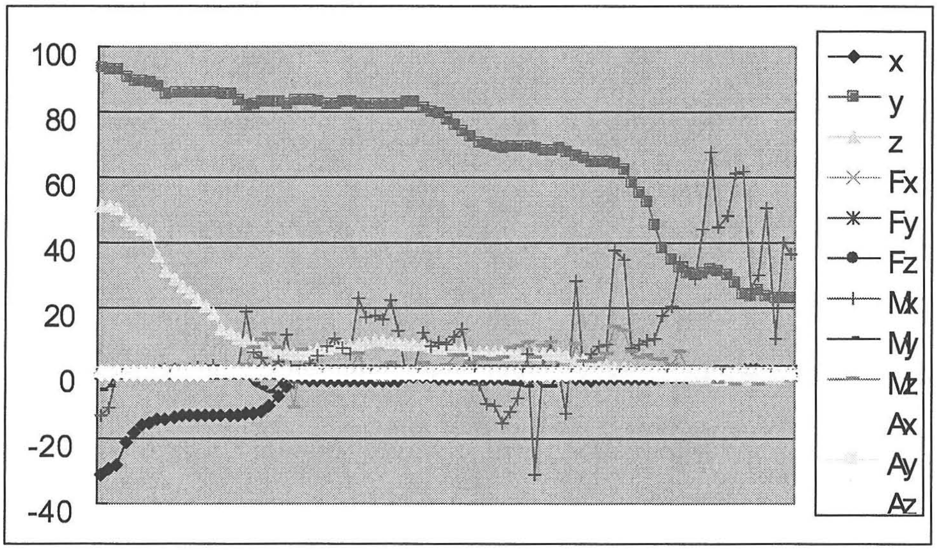


Figure 13 PointShell Method

6. Conclusion

The work conducted to stabilize the virtual manipulation of the peg-in-hole insertion in a virtual environment using a 6 DOF haptic device was reported. A more thorough and precise haptic rendered model using modified PointShell algorithm, modified TriPolyMesh algorithm and dual-gstCylinder algorithm have stabilized the virtual insertion process and has removed the oscillation observed in the system when 3 DOF haptic device was used.

7. References

[1] Y. Chen, and F. Naghdy, "Skill Acquisition in Transfer of Manipulation Skills from Human to Machine through a Haptic Virtual Environment," 2002 IEEE International Conference on Industrial Technology (ICIT) "Productivity Reincarnation through Robotics & Automation", 11-14 December 2002, Bangkok, Thailand, pp.337-342

[2] Y. Chen, and F. Naghdy, "Teaching Manipulation Skills to a Robot through a haptic Rendered Virtual Environment," Advanced Manufacturing Systems (IJAMS), Vol 1, No. 1, p89-106, June 2002.

[3] Grigore C. Burdea, "Haptics Issues in Virtual Environments," Computer Graphics International (CGI'00), June 19 - 24, 2000 Geneva, Switzerland, pp. 295-302,

[4] SensAble Technologies – Haptics Research. [Online], available, http://www.sensable.com/haptics/

[5] SensAble Technologies, "GHOST SDK Programmer's Guide," Cambridge MA, 1999.

[6] Grigore C. Burdea, "Keynote Address: Haptics Feedback for Virtual Reality," Proceedings of International Workshop on Virtual prototyping, Laval, France, pp. 87-96, May 1999.

[7] Aimee Potts, "Phantom-Based Haptic Interaction," [Online], available, http://mrs.umn.edu/~lopezdr/seminar/spring2000/potts.pdf

[8] Allison M. Okamura, "Literature Survey of Haptic Rendering, Collision Detection, and Object Modeling," Dextrous Manipulation Lab Technical Report, August 27, 1998.

[9] John M. Hollerbach, Virtual Environment Rendering Book Chapter Draft for Human and Machine Haptics, Cutkosky, et al. Eds., 1998.

[10] Jason P. Fritz, "AHaptic Object-Oriented Texture Rendering System", Proceedings of the 1997 IEEE Virtual Reality Annual International Symposium, Albuquerque, NM, 1997, p.210.

[11] M. Renz, C. Preusche, M. Pötke, H.-P. Kriegel, G. Hirzinger, "Stable Haptic Interaction with Virtual Environments Using an Adapted Voxmap-PointShell Algorithm". Proceedings of the Eurohaptics Conference, Birmingham, UK, 2001